|  |  |
| --- | --- |
| **EX NO :1(a)** | **Simple Linear Regression** |
| **DATE:** 19/07/2022 | **URK20AI1061** |

**AIM:**

The purpose of this expriment is to predict solar radiation through simple linear regression.

**DESCRIPTION:**

### Linear Regression is a machine learning algorithm based on supervised learning. Linear Regression is a machine learning algorithm based on supervised learning. It is mostly used for finding out the relationship between variables and forecasting. Linear regression is a machine learning technique that predicts a dependent variable value based on a given independent variable value.

**ALGORITHM:**

**STEP 1:** Start

**STEP 2**: Numpy, pandas, and matplotlib must be imported

**STEP 3:** The dataset should be analyzed

**STEP 4:**  From linear\_model, import the LinearRegession model

**STEP 5:** X\_train and Y\_train data must be passed

**STEP 6:** Predict the values of y\_pred

**STEP 7:** Stop

**CODE:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

x = pd.read\_csv('https://github.com/joshuacecil/Datasets/raw/main/Training%20Data%20for%20Linear%20Regression%20-%20Training%20Data%20for%20Linear%20Regression.csv')

X\_train=x['Max Temp']

r=np.array(X\_train)

X\_train=r.reshape(-1,1)

y\_train=x['Solar Radiation']

r=np.array(y\_train)

y\_train=r.reshape(-1,1)

li=LinearRegression()

li.fit(X\_train,y\_train)

tr=pd.read\_csv('https://github.com/joshuacecil/Datasets/raw/main/Testing%20Data%20for%20Linear%20Regression%20-%20Testing%20Data%20for%20Linear%20Regression.csv')

x\_test=tr['Max Temp']

r=np.array(x\_test)

x\_test=r.reshape(-1,1)

y\_test=tr['Solar Radiation']

r=np.array(y\_test)

y\_test=r.reshape(-1,1)

y\_pred=li.predict(x\_test)

y\_pred.shape

print(r2\_score(y\_test,y\_pred))

plt.scatter(x\_test,y\_test,marker='^',label='Actual')

plt.scatter(x\_test,y\_pred,marker = 's',label='Predict')

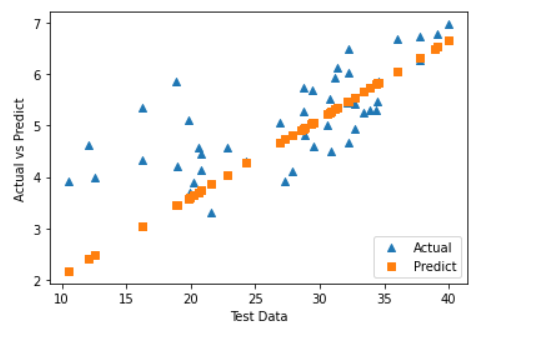
plt.xlabel('Test Data')

plt.ylabel('Actual vs Predict')

plt.legend(loc='lower right')

plt.show()

**OUTPUT:**

****